香 港 中 文 大 學
The Chinese University of Hong Kong

CENG5030

Part 2-3: CNN Inaccurate Speedup-1 —- Overview

Bei Yu

(Latest update: March 25, 2019)

Spring 2019

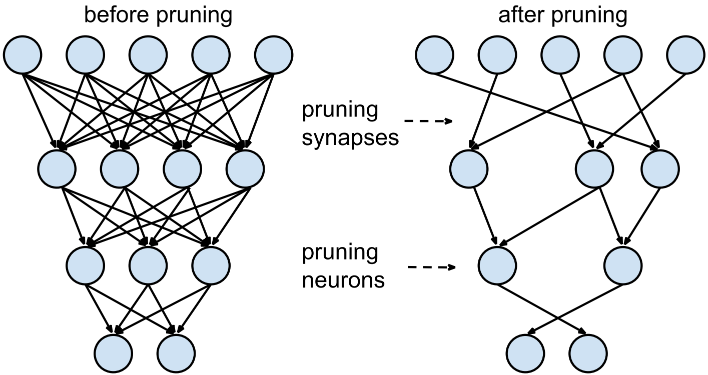
These slides contain/adapt materials developed by

- Song Han, Jeff Pool, et al. (2015). "Learning both weights and connections for efficient neural network". In: *Proc. NIPS*, pp. 1135–1143
- Song Han, Huizi Mao, and William J. Dally (2016). "Deep Compression: Compressing deep neural networks with pruning, trained quantization and huffman coding". In: *Proc. ICLR*
- Song Han, Xingyu Liu, et al. (2016). "EIE: efficient inference engine on compressed deep neural network". In: *Proc. ISCA*, pp. 243–254

Learning both Weights and Connections for Efficient Neural Networks

Han et al.
NIPS 2015

Pruning Neural Networks



Pruning Happens in Human Brain

50 Trillion Synapses

1000 Trillion Synapses

500 Trillion Synapses

Newborn

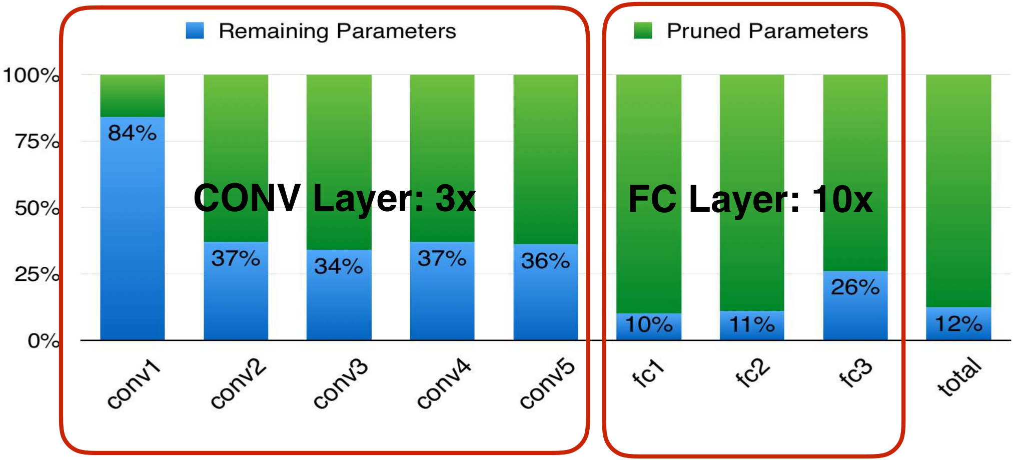
1 year old

Adolescent

Christopher A Walsh. Peter Huttenlocher (1931-2013). Nature, 502(7470):172–172, 2013.

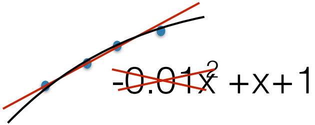
[Han et al. NIPS'15]

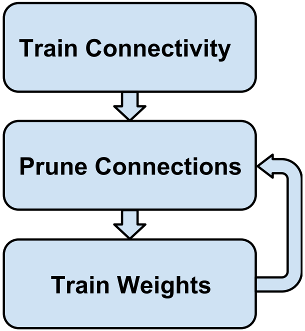
Pruning AlexNet



[Han et al. NIPS’15]

Pruning Neural Networks

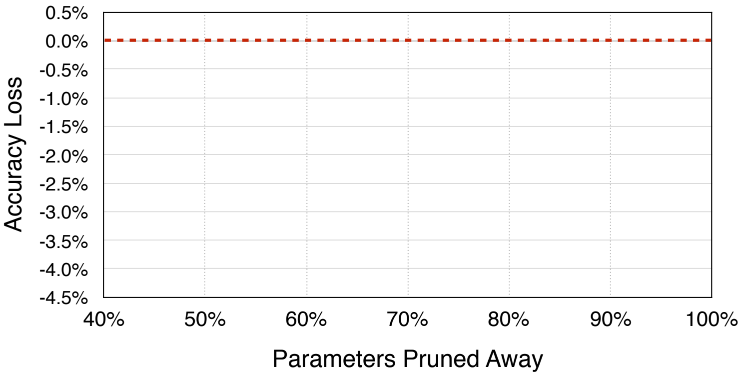




[Han et al. NIPS'15]

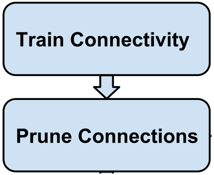
Pruning Neural Networks

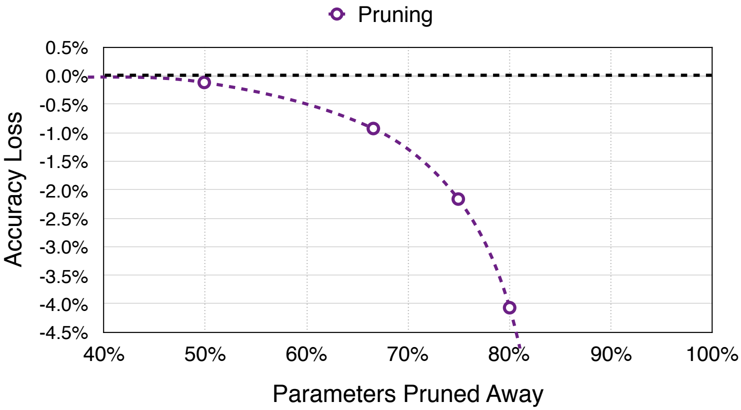
Train Connectivity



[Han et al. NIPS'15]

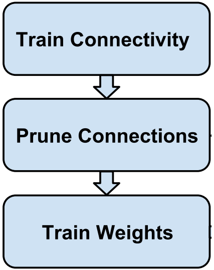
Pruning Neural Networks

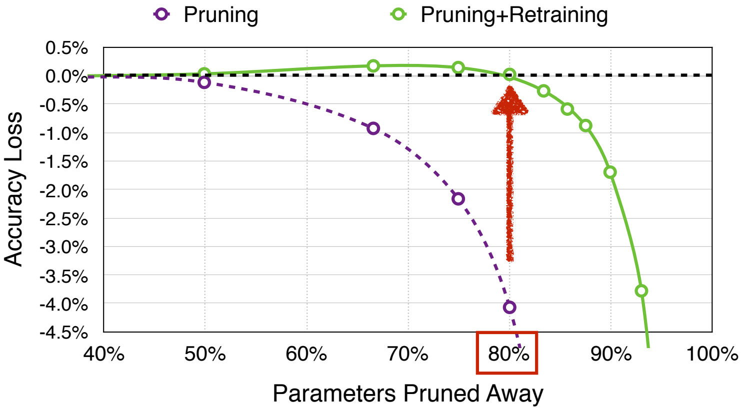




[Han et al. NIPS'15]

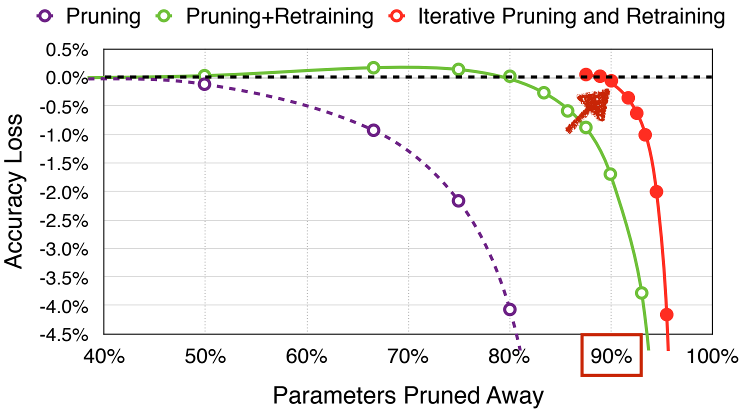
Retrain to Recover Accuracy





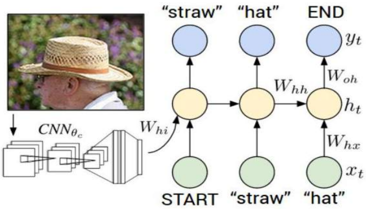
[Han et al. NIPS'15]

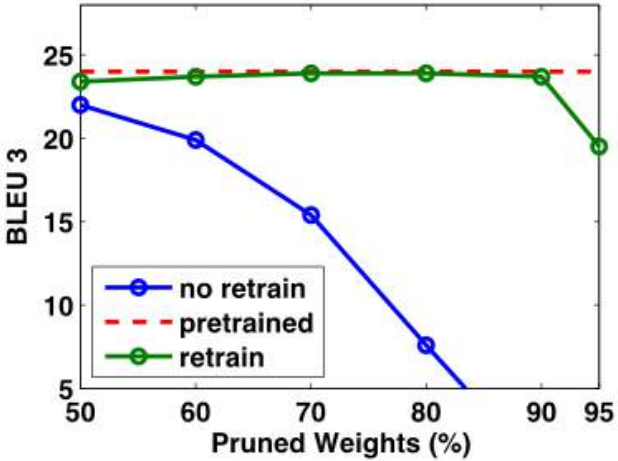
Iteratively Retrain to Recover Accuracy



[Han et al. NIPS'15]

Pruning RNN and LSTM





*Karpathy et al "Deep Visual-Semantic Alignments for Generating Image Descriptions"

[Han et al. NIPS'15]

Pruning RNN and LSTM

90%

- **Original**: a basketball player in a white uniform is playing with a ball
- **Pruned 90%**: a basketball player in a white uniform is playing with a basketball

90%

- **Original** : a brown dog is running through a grassy field
- **Pruned 90%**: a brown dog is running through a grassy area

90%

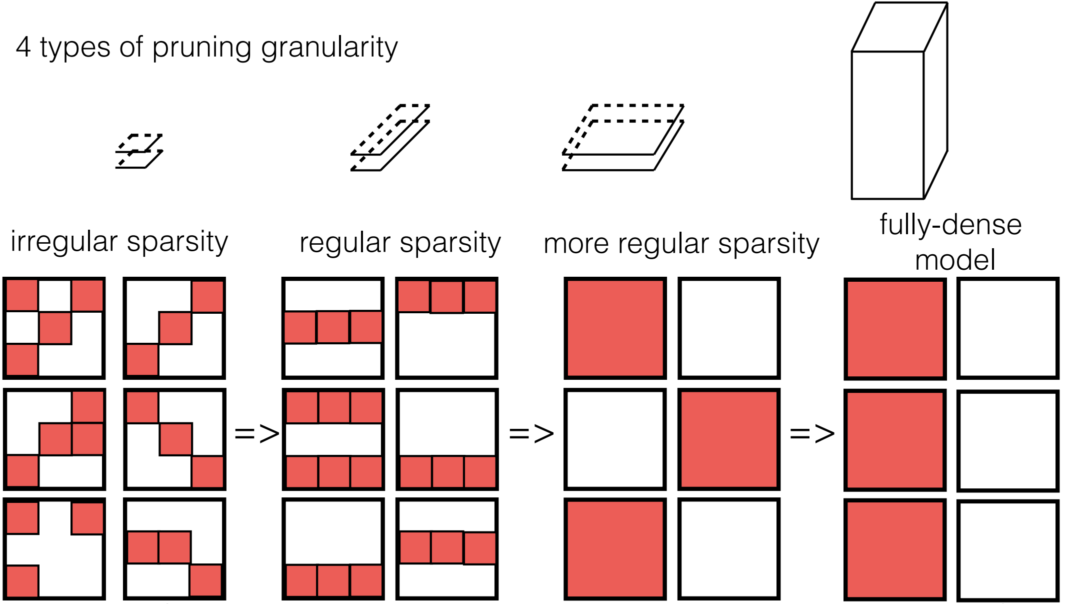
- **Original** : a man is riding a surfboard on a wave
- **Pruned 90%**: a man in a wetsuit is riding a wave on a beach

95%

- **Original** : a soccer player in red is running in the field
- **Pruned <u>95%</u>**: a man in a red shirt and black and white black shirt is running through a field

Exploring the Granularity of Sparsity that is Hardware-friendly

4 types of pruning granularity



[Han et al, NIPS'15] [Molchanov et al, ICLR'17]

Deep Compression: Compressing Deep Neural Networks with Pruning, Trained Quantization and Huffman Coding

Han et al.
ICLR 2016
Best Paper

[Han et al. ICLR’16]

Trained Quantization

~~2.09, 2.12, 1.92, 1.87~~

↓

2.0

[Han et al. ICLR’16]

Trained Quantization

weights
(32 bit float)

2.09	-0.98	1.48	0.09
0.05	-0.14	-1.08	2.12
-0.91	1.92	0	-1.03
1.87	0	1.53	1.49

cluster ⇨

cluster index
(2 bit uint)

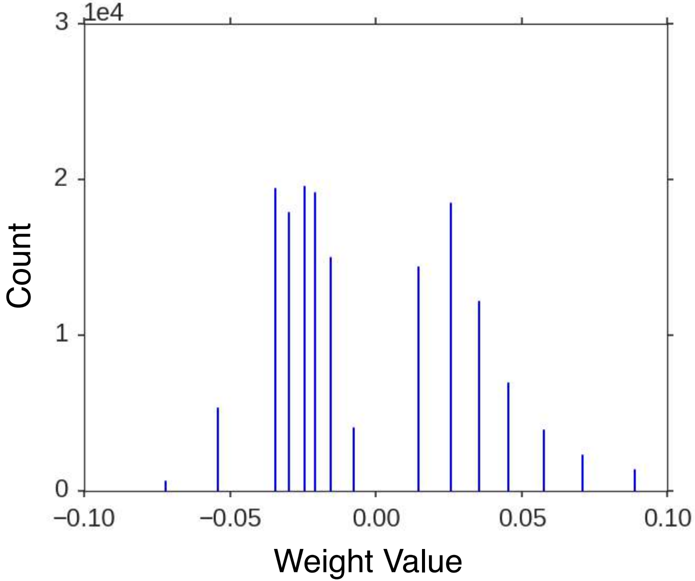
3	0	2	1
1	1	0	3
0	3	1	0
3	1	2	2

centroids

3:	2.00
2:	1.50
1:	0.00
0:	-1.00

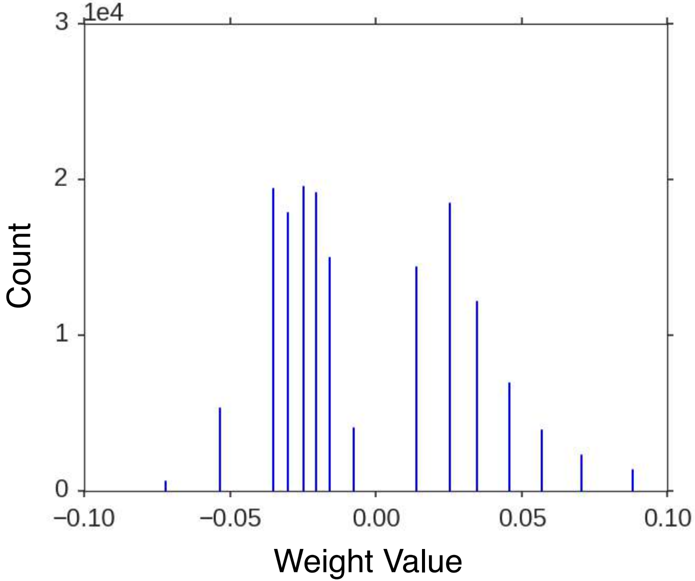
[Han et al. ICLR'16]

After Trained Quantization: Discrete Weight



[Han et al. ICLR'16]

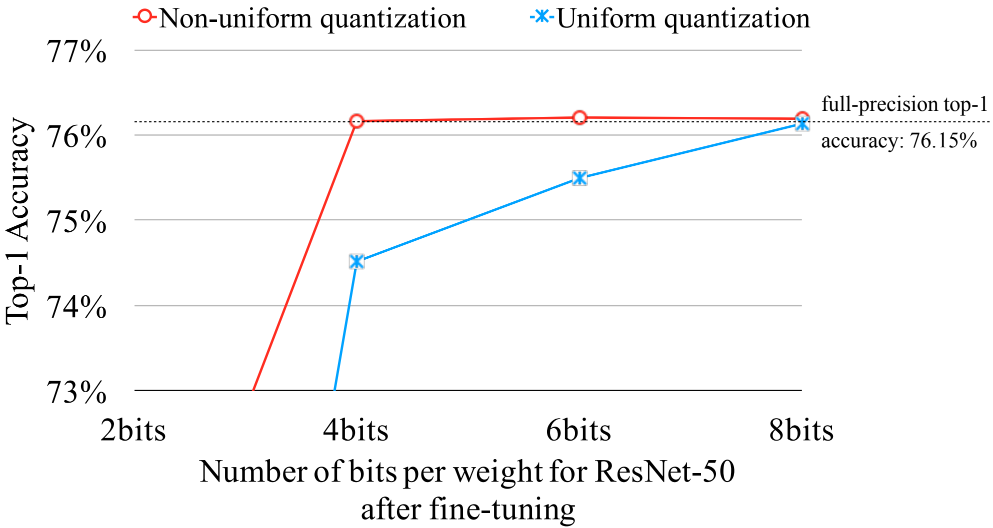
After Trained Quantization: Discrete Weight after Training



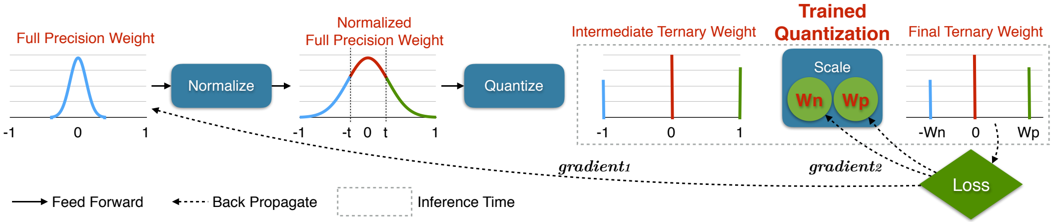
How Many Bits do We Need?

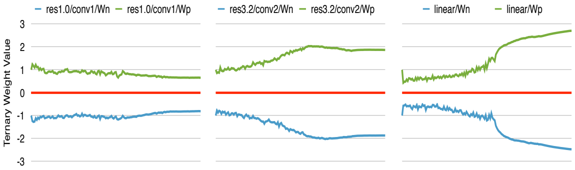
[Han et al. ICLR’16]

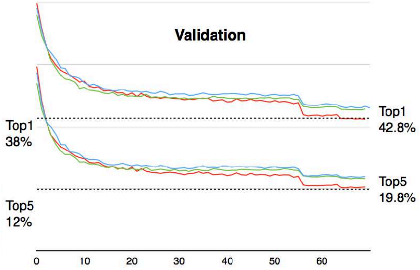
How Many Bits do We Need?



More Aggressive Compression: Ternary Quantization







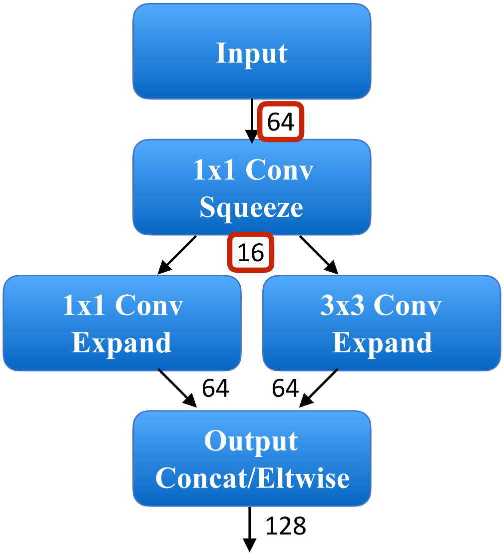
[Han et al. ICLR'16]

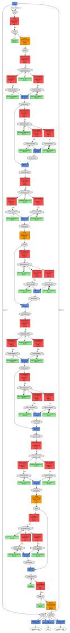
Results: Compression Ratio

Network	Original Size	Compressed Size	Compression Ratio	Original Accuracy	Compressed Accuracy
LeNet-300	1070KB →	27KB	**40x**	98.36% →	98.42%
LeNet-5	1720KB →	44KB	**39x**	99.20% →	99.26%
AlexNet	240MB →	6.9MB	**35x**	80.27% →	80.30%
VGGNet	550MB →	11.3MB	**49x**	88.68% →	89.09%
Inception-V3	91MB →	4.2MB	**22x**	93.56% →	93.67%
ResNet-50	97MB →	5.8MB	**17x**	92.87% →	93.04%

Can we make compact models to begin with?

SqueezeNet



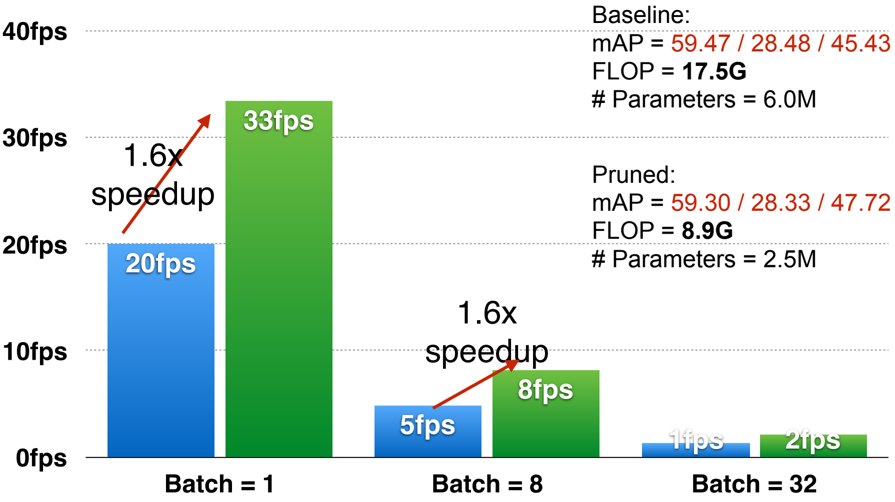


Iandola et al, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size", arXiv 2016

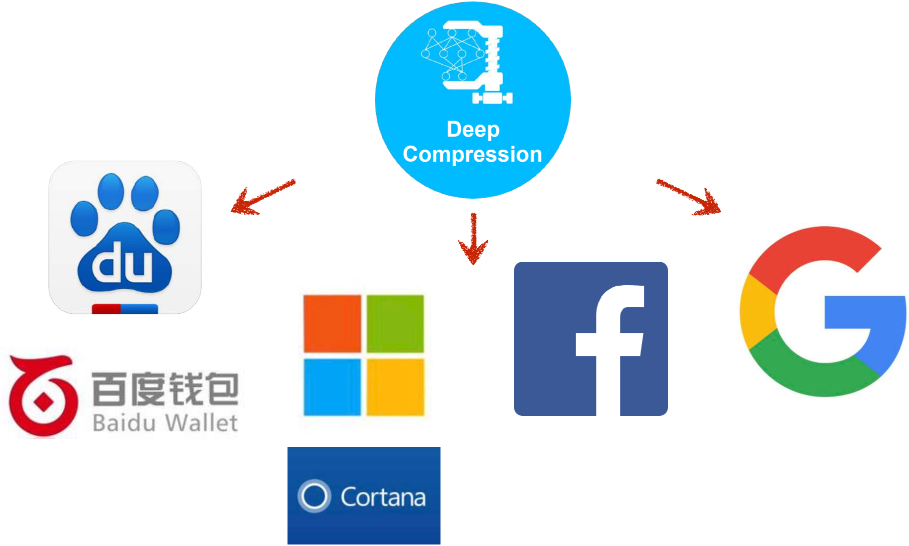
Compressing SqueezeNet

Network	Approach	Size	Ratio	Top-1 Accuracy	Top-5 Accuracy
AlexNet	-	240MB	**1x**	57.2%	80.3%
AlexNet	SVD	48MB	**5x**	56.0%	79.4%
AlexNet	Deep Compression	6.9MB	**35x**	57.2%	80.3%
SqueezeNet	-	4.8MB	**50x**	57.5%	80.3%
SqueezeNet	Deep Compression	0.47MB	$\boxed{\textbf{510x}}$	57.5%	80.3%

Results: Speedup



Deep Compression Applied to Industry



EIE: Efficient Inference Engine on Compressed Deep Neural Network

Han et al.
ISCA 2016

Deep Learning Accelerators

- First Wave: Compute (Neu Flow)
- Second Wave: Memory (Diannao family)
- Third Wave: Algorithm / Hardware Co-Design (EIE)

Google TPU: "This unit is designed for dense matrices. Sparse architectural support was omitted for time-to-deploy reasons. Sparsity will have high priority in future designs"

[Han et al. ISCA'16]

EIE: the First DNN Accelerator for Sparse, Compressed Model

$0 * A = 0$	$W * 0 = 0$	~~2.09, 1.92~~=> 2
Sparse Weight	**Sparse Activation**	**Weight Sharing**
90% *static* sparsity	70% *dynamic* sparsity	4-bit weights

10x less computation

5x less memory footprint

3x less computation

8x less memory footprint

EIE: Parallelization on Sparsity

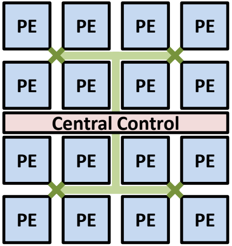
$$\vec{\boldsymbol{a}} \begin{pmatrix} 0 & \boldsymbol{a_1} & 0 & a_3 \end{pmatrix}$$

$$\times$$

$$\left(\begin{array}{c:c:c:c:} w_{0,0} & \boldsymbol{w_{0,1}} & 0 & w_{0,3} \\ 0 & \boldsymbol{0} & w_{1,2} & 0 \\ 0 & \boldsymbol{w_{2,1}} & 0 & w_{2,3} \\ 0 & \boldsymbol{0} & 0 & 0 \\ 0 & \boldsymbol{0} & w_{4,2} & w_{4,3} \\ w_{5,0} & \boldsymbol{0} & 0 & 0 \\ 0 & \boldsymbol{0} & 0 & w_{6,3} \\ 0 & \boldsymbol{w_{7,1}} & 0 & 0 \end{array}\right) = \begin{pmatrix} b_0 \\ b_1 \\ -b_2 \\ b_3 \\ -b_4 \\ b_5 \\ b_6 \\ -b_7 \end{pmatrix} \overset{ReLU}{\Rightarrow} \vec{\boldsymbol{b}} \begin{pmatrix} b_0 \\ b_1 \\ 0 \\ b_3 \\ 0 \\ b_5 \\ b_6 \\ 0 \end{pmatrix}$$

[Han et al. ISCA'16]

EIE: Parallelization on Sparsity



$$\vec{a}\ \begin{pmatrix} 0 & \boldsymbol{a_1} & 0 & a_3 \end{pmatrix}$$

$$\times$$

$$\begin{array}{c} PE0 \\ PE1 \\ PE2 \\ PE3 \\ \\ \\ \\ \\ \end{array}\begin{pmatrix} w_{0,0} & \boldsymbol{w_{0,1}} & 0 & w_{0,3} \\ 0 & \mathbf{0} & w_{1,2} & 0 \\ 0 & \boldsymbol{w_{2,1}} & 0 & w_{2,3} \\ 0 & \mathbf{0} & 0 & 0 \\ 0 & \mathbf{0} & w_{4,2} & w_{4,3} \\ w_{5,0} & \mathbf{0} & 0 & 0 \\ 0 & \mathbf{0} & 0 & w_{6,3} \\ 0 & \boldsymbol{w_{7,1}} & 0 & 0 \end{pmatrix} = \begin{pmatrix} b_0 \\ b_1 \\ -b_2 \\ b_3 \\ -b_4 \\ b_5 \\ b_6 \\ -b_7 \end{pmatrix} \overset{ReLU}{\Rightarrow} \vec{b}\begin{pmatrix} b_0 \\ b_1 \\ 0 \\ b_3 \\ 0 \\ b_5 \\ b_6 \\ 0 \end{pmatrix}$$

[Han et al. ISCA'16]

Dataflow

$$\vec{a} \; \begin{pmatrix} 0 & \boldsymbol{a_1} & 0 & a_3 \end{pmatrix}$$

$$\times$$

$$\begin{array}{c} PE0 \\ PE1 \\ PE2 \\ PE3 \\ \\ \\ \\ \\ \end{array} \begin{pmatrix} w_{0,0} & \boldsymbol{w_{0,1}} & 0 & w_{0,3} \\ 0 & \mathbf{0} & w_{1,2} & 0 \\ 0 & \boldsymbol{w_{2,1}} & 0 & w_{2,3} \\ 0 & \mathbf{0} & 0 & 0 \\ 0 & \mathbf{0} & w_{4,2} & w_{4,3} \\ w_{5,0} & \mathbf{0} & 0 & 0 \\ 0 & \mathbf{0} & 0 & w_{6,3} \\ 0 & \boldsymbol{w_{7,1}} & 0 & 0 \end{pmatrix} = \begin{pmatrix} \boxed{b_0} \\ \boxed{b_1} \\ \boxed{-b_2} \\ \boxed{b_3} \\ \boxed{-b_4} \\ \boxed{b_5} \\ \boxed{b_6} \\ \boxed{-b_7} \end{pmatrix} \overset{ReLU}{\Rightarrow} \vec{b} \begin{pmatrix} b_0 \\ b_1 \\ 0 \\ b_3 \\ 0 \\ b_5 \\ b_6 \\ 0 \end{pmatrix}$$

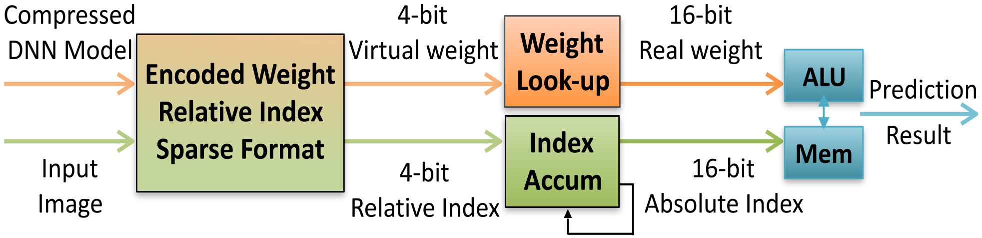
rule of thumb:

0 * A = 0 W * 0 = 0

[Han et al. ISCA'16]

EIE Architecture

Weight decode

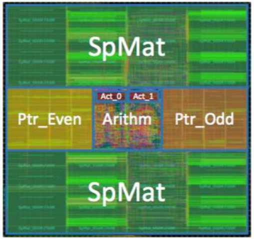


Address Accumulate

rule of thumb: 0 * A = 0 W * 0 = 0 ~~2.09, 1.92~~=> 2

[Han et al. ISCA'16]

Post Layout Result of EIE

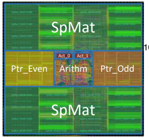


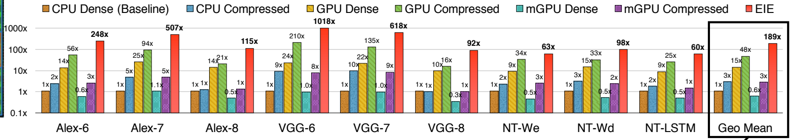
Technology	40 nm
# PEs	64
on-chip SRAM	8 MB
Max Model Size	84 Million
Static Sparsity	10x
Dynamic Sparsity	3x
Quantization	4-bit
ALU Width	16-bit
Area	40.8 mm^2
MxV Throughput	81,967 layers/s
Power	586 mW

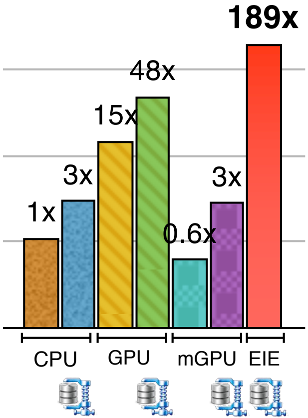
1. **Post layout result**
2. **Throughput measured on AlexNet FC-7**

[Han et al. ISCA'16]

Speedup on EIE



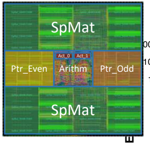


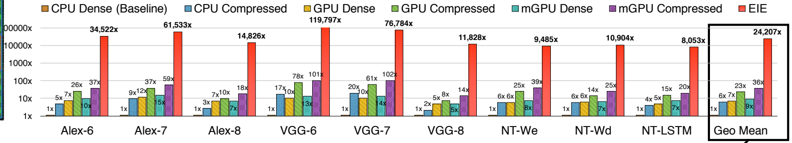


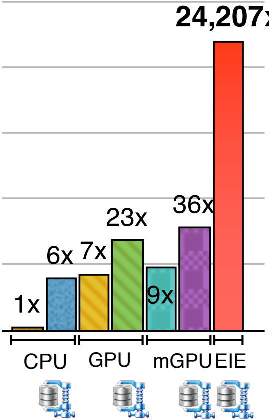
Geo Mean

[Han et al. ISCA'16]

Energy Efficiency on EIE



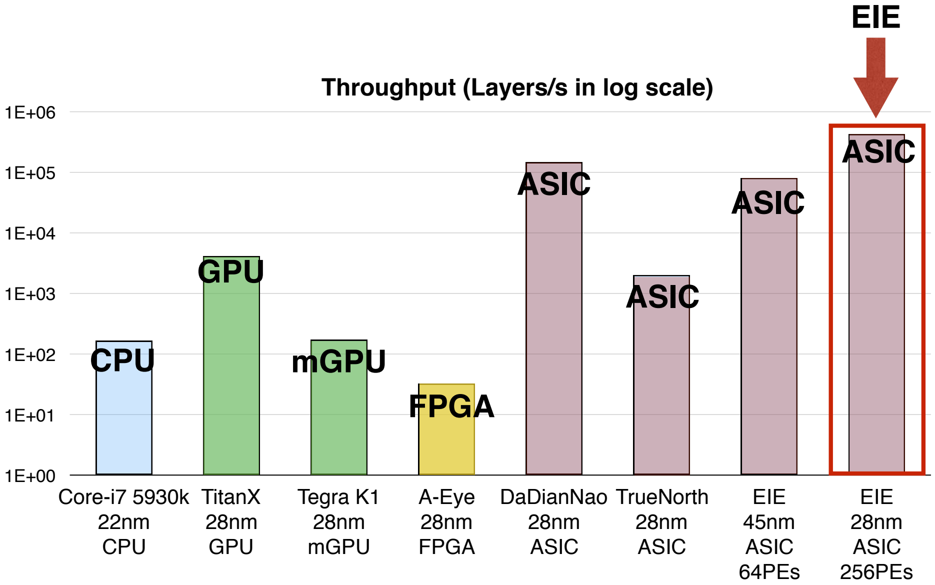




Geo Mean

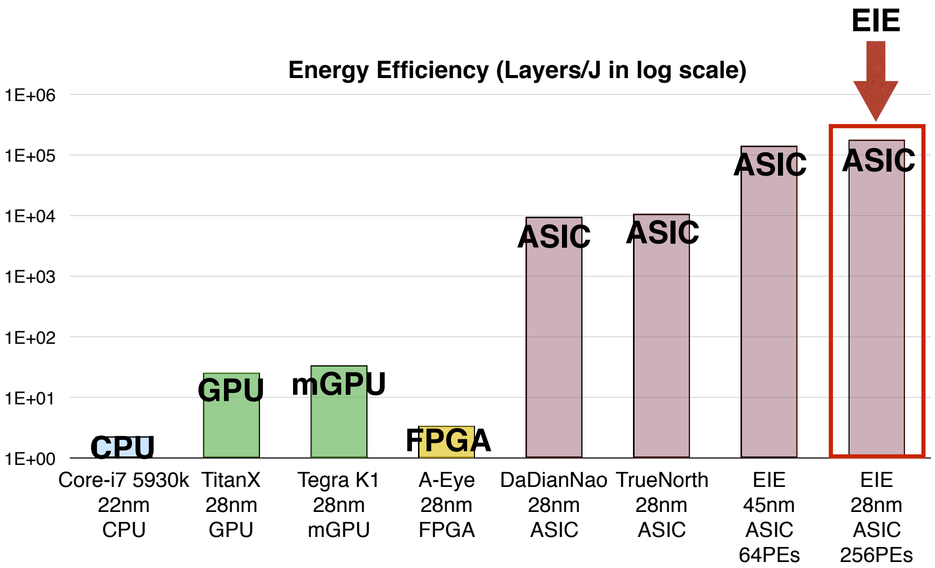
[Han et al. ISCA'16]

Comparison: Throughput



[Han et al. ISCA'16]

Comparison: Energy Efficiency



Further Discussion: Readling List

- Wenlin Chen et al. (2015). "Compressing neural networks with the hashing trick". In: *Proc. ICML*, pp. 2285–2294
- Wei Wen et al. (2016). "Learning structured sparsity in deep neural networks". In: *Proc. NIPS*, pp. 2074–2082
- Huizi Mao et al. (2017). "Exploring the granularity of sparsity in convolutional neural networks". In: *CVPR Workshop*, pp. 13–20
- Zhuang Liu et al. (2017). "Learning efficient convolutional networks through network slimming". In: *Proc. ICCV*, pp. 2736–2744